



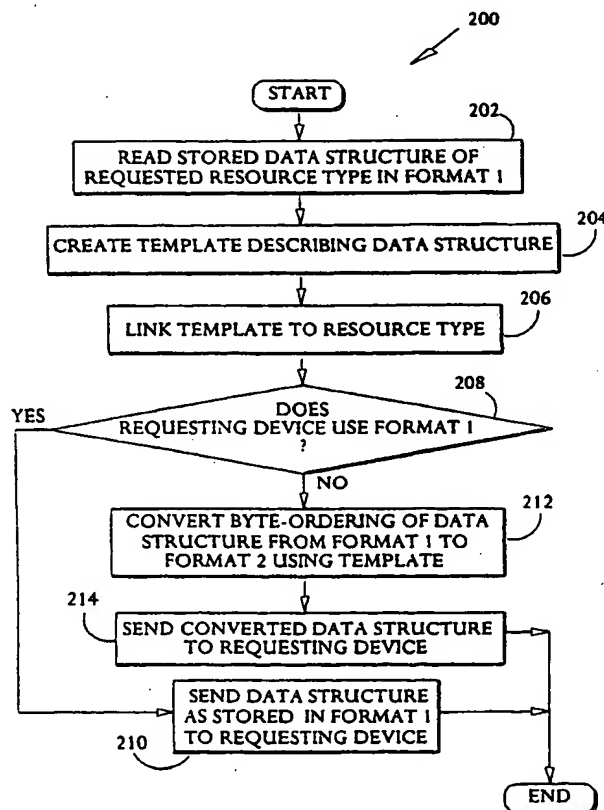
INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 7 : G06F 13/40, 17/30		A1	(11) International Publication Number: WO 00/23902
			(43) International Publication Date: 27 April 2000 (27.04.00)
(21) International Application Number: PCT/US99/23542 (22) International Filing Date: 8 October 1999 (08.10.99) (30) Priority Data: 09/174,072 16 October 1998 (16.10.98) US (71) Applicant: SOFTBOOK PRESS, INC. [US/US]; Suite 200, 1075 Curtis Street, Menlo Park, CA 94025 (US). (72) Inventors: DUGA, Brady; 3342 Cuesta Place, Carlsbad, CA 92009 (US). MARDER, Andrew; 9763 Caminito Doha, San Diego, CA 92131 (US). CONBOY, Garth; 359 Belvedere Street, La Jolla, CA 92037 (US). (74) Agents: BLAKELY, Roger, W. et al.; Blakely, Sokoloff, Taylor & Zafman, 7th floor, 12400 Wilshire Boulevard, Los Angeles, CA 90025-1026 (US).		(81) Designated States: AE, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CR, CU, CZ, DE, DK, DM, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG). Published With international search report.	

(54) Title: DYNAMIC CONVERSION OF BYTE ORDERING FOR USE ON DIFFERENT PROCESSOR PLATFORMS

(57) Abstract

The invention is a method for dynamically converting the byte-ordering of a data structure of a resource type from a first format to a second format, the first format being incompatible with the second format. The method comprises the following steps: (a) creating a template which corresponds to the data structure of the resource type, the template having a structure in a third format which is compatible with the first and second formats; (b) linking the template to the resource type; and (c) converting automatically the byte-ordering of the data structure of the resource type from the first format to the second format using the template.



FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece			TR	Turkey
BG	Bulgaria	HU	Hungary	ML	Mali	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MN	Mongolia	UA	Ukraine
BR	Brazil	IL	Israel	MR	Mauritania	UG	Uganda
BY	Belarus	IS	Iceland	MW	Malawi	US	United States of America
CA	Canada	IT	Italy	MX	Mexico	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NE	Niger	VN	Viet Nam
CG	Congo	KE	Kenya	NL	Netherlands	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NO	Norway	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	NZ	New Zealand		
CM	Cameroon		Republic of Korea	PL	Poland		
CN	China	KR	Republic of Korea	PT	Portugal		
CU	Cuba	KZ	Kazakstan	RO	Romania		
CZ	Czech Republic	LC	Saint Lucia	RU	Russian Federation		
DE	Germany	LI	Liechtenstein	SD	Sudan		
DK	Denmark	LK	Sri Lanka	SE	Sweden		
EE	Estonia	LR	Liberia	SG	Singapore		

DYNAMIC CONVERSION OF BYTE ORDERING FOR USE ON DIFFERENT PROCESSOR PLATFORMS

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention relates generally to a distribution system of digital contents, i.e., digitally encoded published materials. More particularly, the invention relates to a method for dynamically converting the byte-ordering of a data structure in a digital content from one format to a different format.

2. Description of Related Art

Advances in computer and communication technology have provided the consumers a convenient and economical means to access information in a variety of media. One particular area of information access is the electronic books. An electronic book is a viewing device that receives printed materials in the form of digital data downloaded from an information network. A user of an electronic book can read downloaded contents of books and printed materials subscribed from a participating bookstore at his or her own convenience without the need to purchase the printed copies of the books.

An electronic book, or any other viewing device, has persistent memory, such as a hard disk or a flash random-access-memory (RAM), to store the downloaded digital contents and constitutes a particular processor platform.

Digital contents stored as binary data in persistent memory of a viewing device must be readable in the native byte-ordering format of the processor of the viewing device. For performance reasons, certain

processors require the byte-ordering of numeric values to be in specific formats. These formats may be different for different types of processors. For example, for use on a Motorola 680x0 chip, the 2-byte hexadecimal number 3AF7 is stored in RAM as: 3A in byte 1, F7 in byte 2. For use on an Intel processor, the same number is stored in RAM as: F7 in byte 1, 3A in byte 2. The Motorola chip format is called big endian format, and the Intel chip format is called little endian format. The conversion of raw numerical data from one format to the other is further complicated by the fact that the byte-swapping must depend on individual data element sizes. Without knowing the individual data element sizes, it is not possible to convert raw binary numerical data from one format to the other. For instance, two 2-byte numbers are indistinguishable from one 4-byte number. Consider the big endian data ABCD where A, B, C and D are bytes. This data may be two 2-byte numbers, namely, AB and CD, or it may be a single 4-byte number ABCD. Since byte swapping must be done for individual numbers, the resulting little endian data are either BADC if the big endian data are two 2-byte numbers, or DCBA if the big endian data is a single 4-byte number.

For optimal performance, binary data should be stored in the endian format of the final target processor platform for which it is created. However, this is not possible if there is more than one target processor platform and the platforms are not compatible. For example, in a digital content distribution system which includes a virtual bookstore and two different groups of viewing devices which run on two incompatible types of processors, a digital content stored in one endian format at the virtual bookstore can be directly used by one group of viewing devices, but must be converted to a different endian format to be used by the other group. Performing conversion of byte-ordering at the viewing device level is time-consuming and prone to

errors, specially when the data structure has a complex combination of data types and contains nested lists of other structures. In addition, a change to a data structure will require every user of that structure to change its conversion.

Therefore, currently, there is a need for an efficient method to dynamically convert the byte-ordering of a data structure from one endian format to another endian format for use on a different processor platform.

SUMMARY OF THE INVENTION

The present invention is a method for dynamically converting the byte-ordering of a data structure of a resource type from a first format to a second format, the first format being incompatible with the second format. The method comprises the following steps: (a) creating a template which corresponds to the data structure of the resource type, the template having a structure in a third format which is compatible with the first and second formats; (b) linking the template to the resource type; and (c) converting automatically the byte-ordering of the data structure of the resource type from the first format to the second format using the template.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 is a diagram illustrating a system in which the present invention can be practiced.

Figure 2 is a flowchart illustrating one embodiment of the present invention.

Figure 3 is a table describing some values of the template used in the present invention.

DETAILED DESCRIPTION OF THE INVENTION

The present invention is a method and system for dynamically converting the byte-ordering of a data structure of a resource type from a first format to a second format which is incompatible with the first format. The method includes the steps of creating a template having a structure which corresponds to the data structure of the resource type, the template structure being in a third format which is compatible with the first and second formats, linking the template structure to the resource type, and automatically converting the byte-ordering of the data structure of the resource type from the first format to the second format using the template structure.

In the following description, for purposes of explanation, numerous details are set forth in order to provide a thorough understanding of the present invention. However, it will be apparent to one skilled in the art that these specific details are not required in order to practice the present invention.

Figure 1 is a diagram illustrating a system 100 in which one embodiment of the invention can be practiced.

Referring to Figure 1, the system 100 comprises: (a) at least one portable electronic book 10 operative to request a digital content from a catalog of distinct digital contents, to receive and display the requested digital content in readable form; (b) an information services system 20 which includes an authentication server 32 for authenticating the identity of the requesting portable electronic book 10 and a copyright protection server 22 for rendering the requested digital content sent to

the requesting portable electronic book 10 readable only by the requesting portable electronic book 10; (c) at least one primary virtual bookstore 40 in electrical communication with the information services system 20, the primary virtual bookstore being a computer-based storefront accessible by the portable electronic book and including the catalog of distinct digital contents; and (d) a repository 50, in electrical communication with the primary virtual bookstore 40, for storing the distinct digital contents listed in the catalog.

The system 100 preferably includes more than one portable electronic book 10, to be commercially viable. This is illustrated in Figure 1 by including the portable electronic books 12 and 14. The system also preferably includes more than one primary virtual bookstore 40, each serving a different set of customers, each customer owning a portable electronic book.

The system 100 can further comprise a secondary virtual bookstore 60 in electrical communication with the information services system 20. In this case, the information services system 20 also includes a directory of virtual bookstores 26 in order to provide the portable electronic book 10 with access to the secondary virtual bookstore 60 and its catalog of digital contents.

The information services system 20 can optionally include a notice board server for sending messages from one of the virtual bookstores, primary or secondary, to a portable electronic book in the system.

The information services system 20 also includes a registration server 24 for keeping track of the portable electronic books that are considered active accounts in the system and for ensuring that each portable electronic book is associated with a primary virtual bookstore

in the system. In the case where the optional notice board server is included in the information services system 20, the registration server 24 also allows each portable electronic book user to define his/her own notice board and document delivery address.

The information services system 20 preferably comprises a centralized bookshelf 30 associated with each portable electronic book 10 in the system. Each centralized bookshelf 30 contains all digital contents requested and owned by the associated portable electronic book 10. Each portable electronic book 10 user can permanently delete any of the owned digital contents from the associated centralized bookshelf 30. Since the centralized bookshelf 30 contains all the digital contents owned by the associated portable electronic book 10, these digital contents may have originated from different virtual bookstores. The centralized bookshelf 30 is a storage extension for the portable electronic book 10. Such storage extension is needed since the portable electronic book 10 has limited non-volatile memory capacity.

The user of the portable electronic book 10 can add marks, such as bookmarks, inking, highlighting and underlining, and annotations on a digital content displayed on the screen of the portable electronic book, then stores this marked digital content in the non-volatile memory of the electronic book 10. The user can also upload this marked digital content to the information services system 20 to store it in the centralized bookshelf 30 associated with the portable electronic book 10, for later retrieval. It is noted that there is no need to upload any unmarked digital content, since it was already stored in the centralized bookshelf 30 at the time it was first requested by the portable electronic book 10.

The information services system 20 further includes an Internet Services Provider (ISP) 34 for providing Internet network access to each portable electronic book in the system.

In the system 100 of Figure 1, a digital content stored in the repository 50 of the virtual bookstore 40 or in the centralized bookshelf 30 may include numerical values which are stored as binary data in an endian format. If the requesting portable electronic book 10 runs on a processor platform which is incompatible with this endian format, then the byte-ordering of the binary data have to be converted to the other endian format.

Binary data are stored in a number of different record structures. Each structure has an arbitrarily complex combination of data types which may include 2-byte and 4-byte integer values, with and without alignment, NULL terminated strings, counted strings (1-byte and 2-byte counts), points (2-byte or 4-byte integer pairs), etc. In addition, these structures may contain nested lists of other structures. The nested lists may be counted lists or indefinite lists. Thus, due to the complexity of data structures, performing conversion of byte-ordering at the electronic book level is time-consuming and prone to errors. In addition, a change to a data structure will require every user of that data structure to change its conversion.

The present invention allows the byte-ordering conversion to be performed dynamically at the server end of the system of Figure 1.

Figure 2 is a flowchart illustrating one embodiment of the present invention.

Upon START, the process 200 reads a data structure of a resource type requested by a device. The data structure is stored in format 1 in a

simple database system of a server (Block 202). This simple database system is usually called the resource file system.

The process then creates a template describing the data structure of the resource type (Block 204). The structure of this template is in a format which is compatible with both format 1 and format 2. The template structure is a series of 1-byte values that represent individual data elements of the data structure of the resource type. This one-byte format is compatible to both endian formats discussed above, and does not require any conversion of the byte ordering.

The process 200 then links the template to the resource type (Block 206). The template is preferably stored in the same resource file as that of the resource type.

When a device requests the data of the resource type, process 200 determines whether the requesting device uses format 1 for byte-ordering (Block 208). If the requesting device uses format 1, then the data of the resource type as stored in the database system is sent to the requesting device (Block 210). Then, the process 200 is terminated.

If the requesting device does not use format 1, then process 200 converts the byte-ordering of the data structure of the resource type from format 1 to format 2 using the template (Block 212). Then, the process 200 is terminated.

One of the most important elements of the invention is the easily expandable template. Since the template structure is in the format of 1-byte values, byte swapping is not required, making it compatible to any platform. In addition, any change to a data structure layout require only a single change in the corresponding template for all users of that data.

Figure 3 shows the description of some template values. The template data can be expanded to include more values.

Each template describes a resource type. For example, a resource type may have the following data structure, where a short is a 2-byte integer, a long is a 4-byte integer, and a char is a single byte:

Short	v1;
Short	v2;
Char	v3;
Long	v4;
Short	v5;

The template for this resource type is the following integer stream, using the byte values from Figure 3:

2,2,1,4,3,2

or, equivalently, as text:

Short, Short, Byte, Short align, Long, Short

In this example, the template has one more value than the number of elements in the structure. The data for this structure may be aligned, that is, the individual data elements may need to start on 2-byte or 4-byte boundaries. In this example, the alignment is for 2 byte boundaries, forcing the "char" to take 2 bytes, but treating the second byte as padding to be disposed, not as data to be swapped.

The following are additional examples of data structures and their corresponding templates:

1. The following long aligned data structure:

```
Short          s1;  
Byte           flags;  
Long point     lp1;
```

corresponds to the following template:

Short, byte, long align, long point

2. The following short aligned structure:

```
Short          s1;  
Char*          str1;  
Long           count; //count for following string  
Char*          str2;
```

corresponds to the following template:

Short, NULL terminated string, short align, long counted string

3. The following short aligned data structure which includes a number of lists:

```
Short          slc;  
{  
    Short      s1;  
    Byte       s2;  
}  
long           llc;  
{  
    long       slc;
```

-11-

```

    {
        short      s3;
        short      s4;
    }
    char*          str;
}
{
    short          s4;
    long           l1;
}

```

corresponds to the following template:

begin short counted list, short, byte, short align, end list, begin long counted list, begin long counted list, short, short, end list, NULL terminated string, short align, end list, begin indefinite list, short, long

4. The following data structure which consists of a simple list:

```

Short          slc;
{
    long        l1;
    short       s1;
}

```

corresponds to the following template:

begin short counted list, long, short, end counted list

An example of such a simple data structure is a 2-element list with values {ABCD, EF}, {UVWX, YZ}. This data structure would be stored in persistent big endian memory as: 02ABCDEFUVWXYZ, where 02 indicates that there are two elements in the list, each element consists of a long integer followed by a short integer. After using the template to convert the byte-ordering of the data, the data will be stored in the memory of a little endian device as: 20DCBAFEXWVUZY.

While certain exemplary embodiments have been described in detail and shown in the accompanying drawings, it is to be understood that such embodiments are merely illustrative of and not restrictive on the broad invention, and that this invention is not to be limited to the specific arrangements and constructions shown and described, since various other modifications may occur to those with ordinary skill in the art.

CLAIMS

What is claimed is:

1. A method for dynamically converting byte-ordering of a data structure of a resource type from a first format to a second format, the first format being incompatible with the second format, the method comprising the steps of:

(a) creating a template corresponding to the data structure of the resource type, the template having a structure in a third format, the third format being compatible with the first and second formats;

(b) linking the template to the resource type; and

(c) converting automatically the byte-ordering of the data structure of the resource type from the first format to the second format using the template.

2. The method of Claim 1 wherein the template is expandable.

3. The method of Claim 1 wherein the template structure comprises one-byte values.

4. The method of Claim 1 wherein the first format is a big endian format and the second format is a little endian format.

5. The method of Claim 1 wherein the template includes a value representing a byte.

6. The method of Claim 1 wherein the template includes a value representing one selected from the list of a byte, a short integer, a long integer, a short align indicator, a long align indicator, a null terminated string, a short counted string, a byte counted string, a short point, a long point, an indefinite list, a start of a short counted list, a start of a long counted list, and an end of a counted list.

7. A system for dynamically converting byte-ordering of a data structure of a resource type from a first format to a second format, the first format being incompatible with the second format, the system comprising:

(a) a server including:

a resource file system having a database for storing the resource type, the resource type being stored in a first format, the resource file system creating a template corresponding to the data structure of the resource type, the template having a template structure in a third format, the third format being compatible with the first and second formats, the resource file system linking the template to the resource type; and

(b) a device in electrical communication with the server, the device requesting the resource type in the second format;

wherein the resource file system automatically converts the byte-ordering of the data structure of the resource type from the first format to the second format using the template.

8. The system of Claim 7 wherein the device is an electronic book.

9. The system of Claim 7 wherein the server is a virtual bookstore.

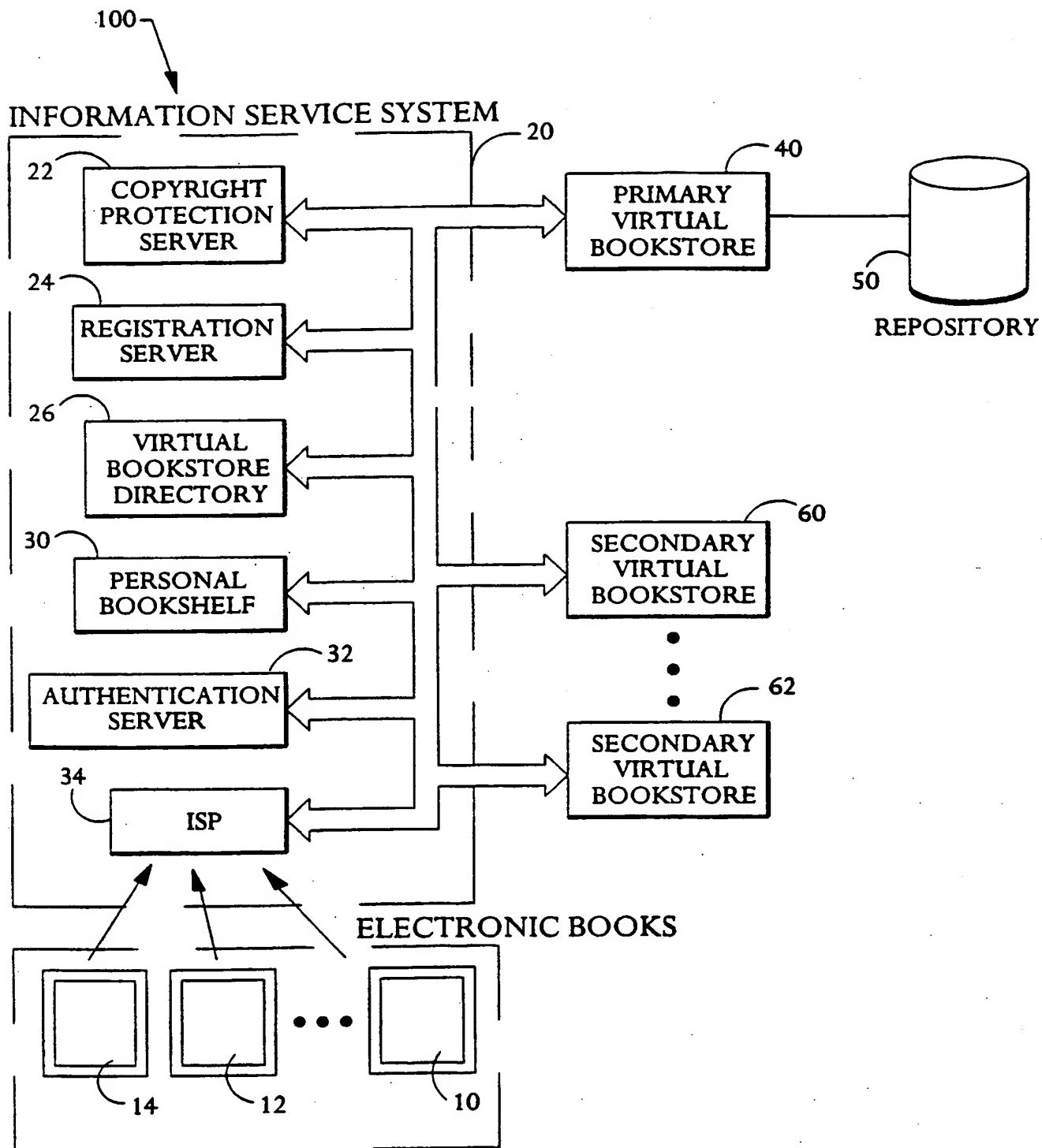


Fig. 1

2/4

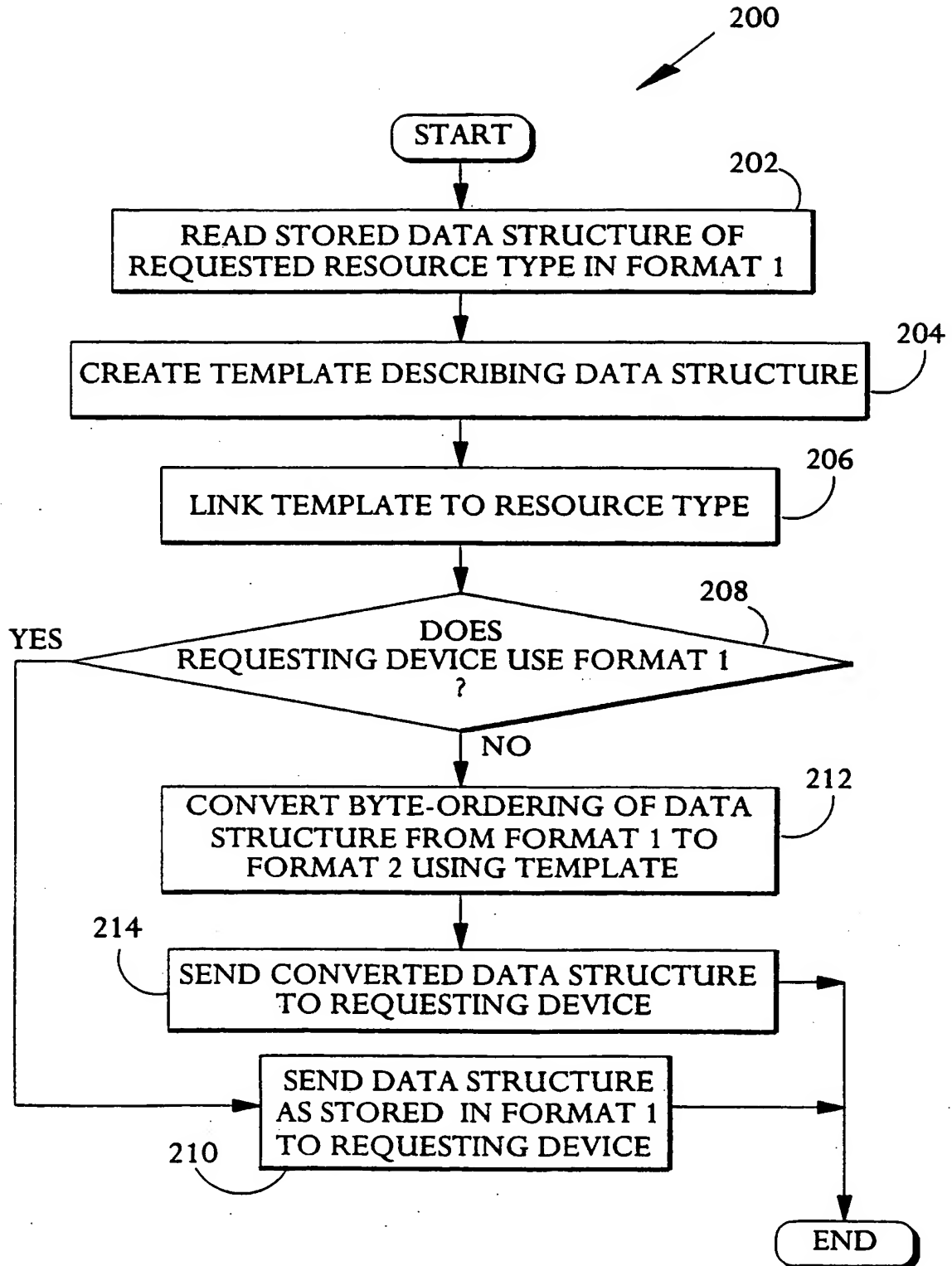


Fig. 2

VALUE	DESCRIPTION
Byte (1)	A single byte - don't perform swapping
Short (2)	A 2 byte integer - swap the next 2 bytes
Long (3)	A 4 byte integer - swap the next 4 bytes
Short Align (4)	If necessary, insert a pad character to align on a 2-byte boundary. Do not perform swapping on this data.
Long Align (5)	If necessary, add enough padding to put the next value on a 4 byte boundary (adds up to 3 pad bytes). Do not perform swapping on this data.
Null terminated string (6)	The following data is a stream of bytes, ending on a NULL (0) byte. Perform no byte swapping.
Short counted string (7)	The following data is a 2-byte integer (count), followed by count bytes. Swap the next 2 bytes, but do not swap the count bytes following.
Byte counted string (8)	The following data is a 1-byte integer (count), followed by count bytes. Do not swap this data.

Fig. 3A

VALUE	DESCRIPTION
Short point (9)	The following 4 bytes is a pair of 2 byte numbers. Swap them as if they were 2 2-byte integers.
Long point (10)	The following 8 bytes is a pair of 4 byte numbers. Swap them as if they were 2 4-byte integers.
Indefinite list (11)	The remaining data in the resource is a list of structures. All elements remaining in the template will describe the structure of a single list element. No data is swapped or skipped.
Begin short counted list (13)	The next 2 bytes are a count for a list. All following data in the template until "End list" is treated as the layout for a single list element. The first 2 bytes (count) are swapped, and then count list elements are swapped as specified by the list element template.
Begin long counted list (14)	Same as a short counted list, but the count is 4 bytes.
End list (12)	Identifies the end of a counted list (either short or long count). Does not correspond to actual data, thus performing no byte swapping.

Fig. 3B

INTERNATIONAL SEARCH REPORT

Int. Application No

PCT/US 99/23542

A. CLASSIFICATION OF SUBJECT MATTER

IPC 7 G06F13/40 G06F17/30

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC 7 G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	PATENT ABSTRACTS OF JAPAN vol. 1995, no. 11, 26 December 1995 (1995-12-26) & JP 07 219823 A (HITACHI), 18 August 1995 (1995-08-18) abstract	1-9
Y	WO 98 06034 A (GENERAL MAGIC) 12 February 1998 (1998-02-12) page 3, line 7 - line 16	1-9
A	US 5 557 790 A (BINGHAM RONALD E ET AL) 17 September 1996 (1996-09-17) claims 1-9	1-9
	-/--	

☒ Further documents are listed in the continuation of box C.

☒ Patent family members are listed in annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search

1 February 2000

Date of mailing of the international search report

07/02/2000

Name and mailing address of the ISA

European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040, Tx. 31 651 epo nl,
Fax: (+31-70) 340-3016

Authorized officer

Gill, S

INTERNATIONAL SEARCH REPORT

International Application No

PCT/US 99/23542

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT		
Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	EP 0 751 655 A (THOMSON) 2 January 1997 (1997-01-02) page 3, column 3, line 26 -page 5, column 7, line 2; figure 1. -----	1-9
A	EP 0 390 611 A (HIUKA SANGYO KK) 3 October 1990 (1990-10-03) page 5, line 51 -page 6, line 2; figures 8,9 -----	8,9

INTERNATIONAL SEARCH REPORT

information on patent family members

Int. onal Application No

PCT/US 99/23542

Patent docum nt cited in search r port	Publication date	Patent family member(s)	Publication date
JP 07219823 A	18-08-1995	NONE	
WO 9806034 A	12-02-1998	AU 3744297 A	25-02-1998
US 5557790 A	17-09-1996	NONE	
EP 751655 A	02-01-1997	US 5778406 A	07-07-1998
		JP 9034817 A	07-02-1997
		SG 40881 A	14-06-1997
EP 0390611 A	03-10-1990	JP 2257364 A	18-10-1990
		US 5239665 A	24-08-1993

THIS PAGE BLANK (USPTO)